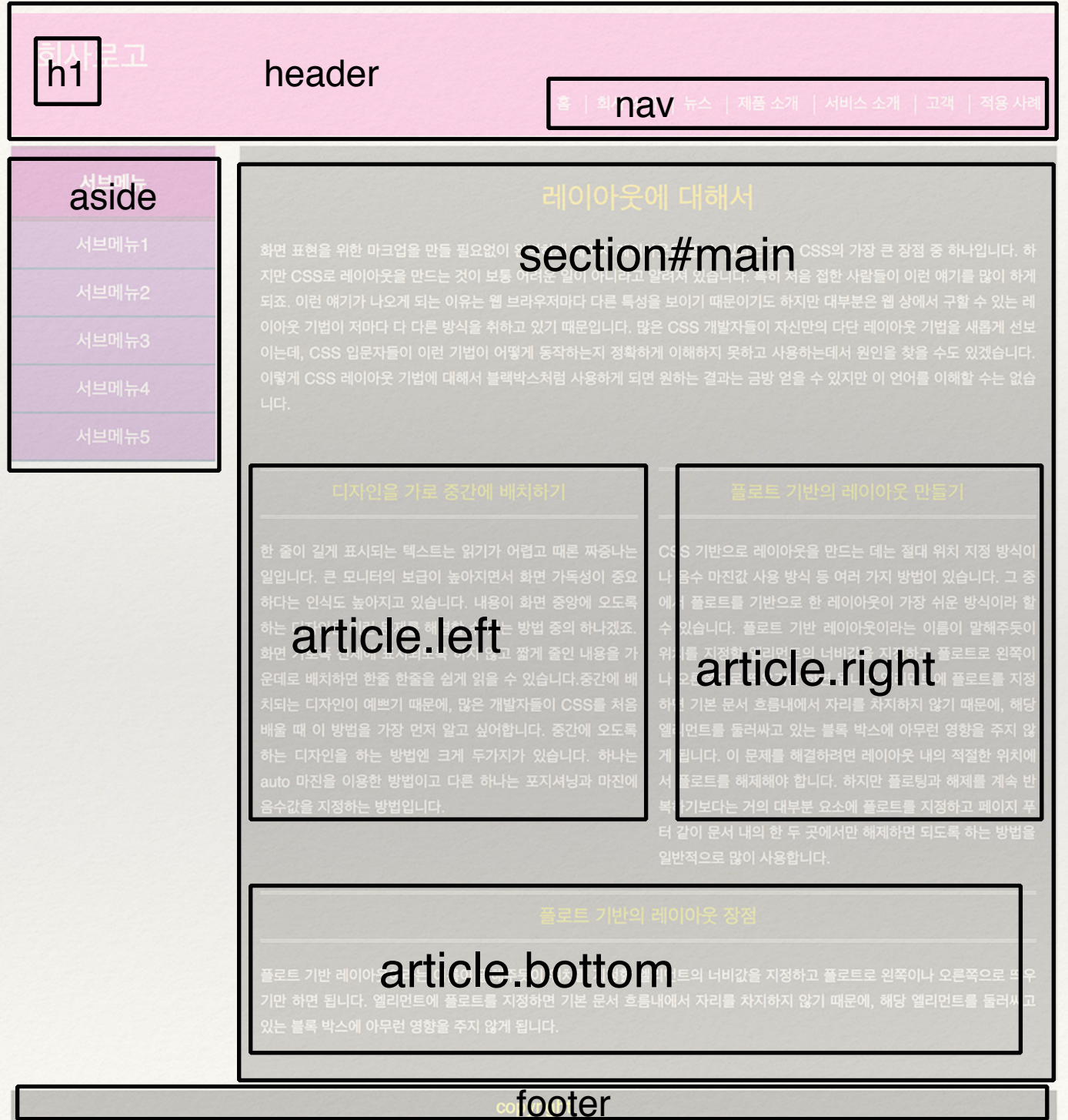


layout01.html

구조



구조

div#headerWrap

div#contents

Header

Section

article.right

article.left

article.bottom

Footer

회사소개
h1

hn* nav>ul>li>a

홈 | 회사 소개 | 뉴스 | 제품 소개 | 서비스 소개 | 고객 | 적용 사례

hn

서브메뉴1

서브메뉴2

서브메뉴3

서브메뉴4

서브메뉴5

ul>li>a

레이아웃에 대해서
hn

화면 표현을 위한 마크업을 만들 필요없이 완벽하게 페이지 레이아웃을 짤 수 있다는 것은 CSS의 가장 큰 장점 중 하나입니다. 하지만 CSS로 레이아웃을 만드는 것이 보통 어려운 일이 아니라고 알려져 있습니다. 특히 처음 접한 사람들이 이런 얘기를 많이 하게 되죠. 이런 얘기가 나오게 되는 이유는 웹 브라우저마다 다른 특성을 보이기 때문이기도 하지만 대부분은 웹 상에서 구할 수 있는 레이아웃 기법이 저마다 다 다른 방식을 취하고 있기 때문입니다. 많은 CSS 개발자들이 자신만의 다단 레이아웃 기법을 새롭게 선보이는데, CSS 입문자들이 이런 기법이 어떻게 동작하는지 정확하게 이해하지 못하고 사용하는데서 원인을 찾을 수도 있겠습니다. 이렇게 CSS 레이아웃 기법에 대해서 블랙박스처럼 사용하게 되면 원하는 결과는 금방 얻을 수 있지만 이 언어를 이해할 수는 없습니다.

디자인을 가로 중간에 배치하기
hn

한 줄이 길게 표시되는 텍스트는 읽기가 어렵고 때론 짜증나는 일입니다. 큰 모니터의 보급이 높아지면서 화면 가독성이 중요하다는 인식도 높아지고 있습니다. 내용이 화면 중앙에 오도록 하는 디자인은 이런 문제를 해결할 수 있는 방법 중의 하나겠지요. 화면 가로 폭 전체에 표시되도록 하지 않고 짧게 줄인 내용을 가운데로 배치하면 한줄 한줄을 쉽게 읽을 수 있습니다.중간에 배치되는 디자인이 예쁘기 때문에, 많은 개발자들이 CSS를 처음 배울 때 이 방법을 가장 먼저 알고 싶어합니다. 중간에 오도록 하는 디자인을 하는 방법엔 크게 두가지가 있습니다. 하나는 auto 마진을 이용한 방법이고 다른 하나는 포지셔닝과 마진에 음수값을 지정하는 방법입니다.

플로트 기반의 레이아웃 만들기
hn

CSS 기반으로 레이아웃을 만드는 데는 절대 위치 지정 방식이나 음수 마진값 사용 방식 등 여러 가지 방법이 있습니다. 그 중에서 플로트를 기반으로 한 레이아웃이 가장 쉬운 방식이라 할 수 있습니다. 플로트 기반 레이아웃이라는 이름이 말해주듯이 위치를 지정할 엘리먼트의 너비값을 지정하고 플로트로 왼쪽이나 오른쪽으로 띄우거나 하면 됩니다.엘리먼트에 플로트를 지정하면 기본 문서 흐름내에서 자리를 차지하지 않기 때문에, 해당 엘리먼트를 둘러싸고 있는 블록 박스에 아무런 영향을 주지 않게 됩니다. 이 문제를 해결하려면 레이아웃 내의 적절한 위치에서 플로트를 해제해야 합니다. 하지만 플로팅과 해제를 계속 반복하기보다는 거의 대부분 요소에 플로트를 지정하고 페이지 폭이 같이 문서 내의 한 두 곳에서만 해제하면 되도록 하는 방법을 일반적으로 많이 사용합니다.

플로트 기반의 레이아웃 장점
hn

플로트 기반 레이아웃이라는 이름이 말해주듯이 위치를 지정할 엘리먼트의 너비값을 지정하고 플로트로 왼쪽이나 오른쪽으로 띄우기만 하면 됩니다. 엘리먼트에 플로트를 지정하면 기본 문서 흐름내에서 자리를 차지하지 않기 때문에, 해당 엘리먼트를 둘러싸고 있는 블록 박스에 아무런 영향을 주지 않게 됩니다.

copyright

준비하기

❖ 스타일연결하기

1. reset.css연결하기
2. layout.css연결하기

❖ 공통속성 만들기

4. .hideClass
5. body에 공통부분넣기

Layout 만들기

A

#headerwrap

크기	가로	Body
배경		

header

크기	가로	A
여백	아래	B

#contents

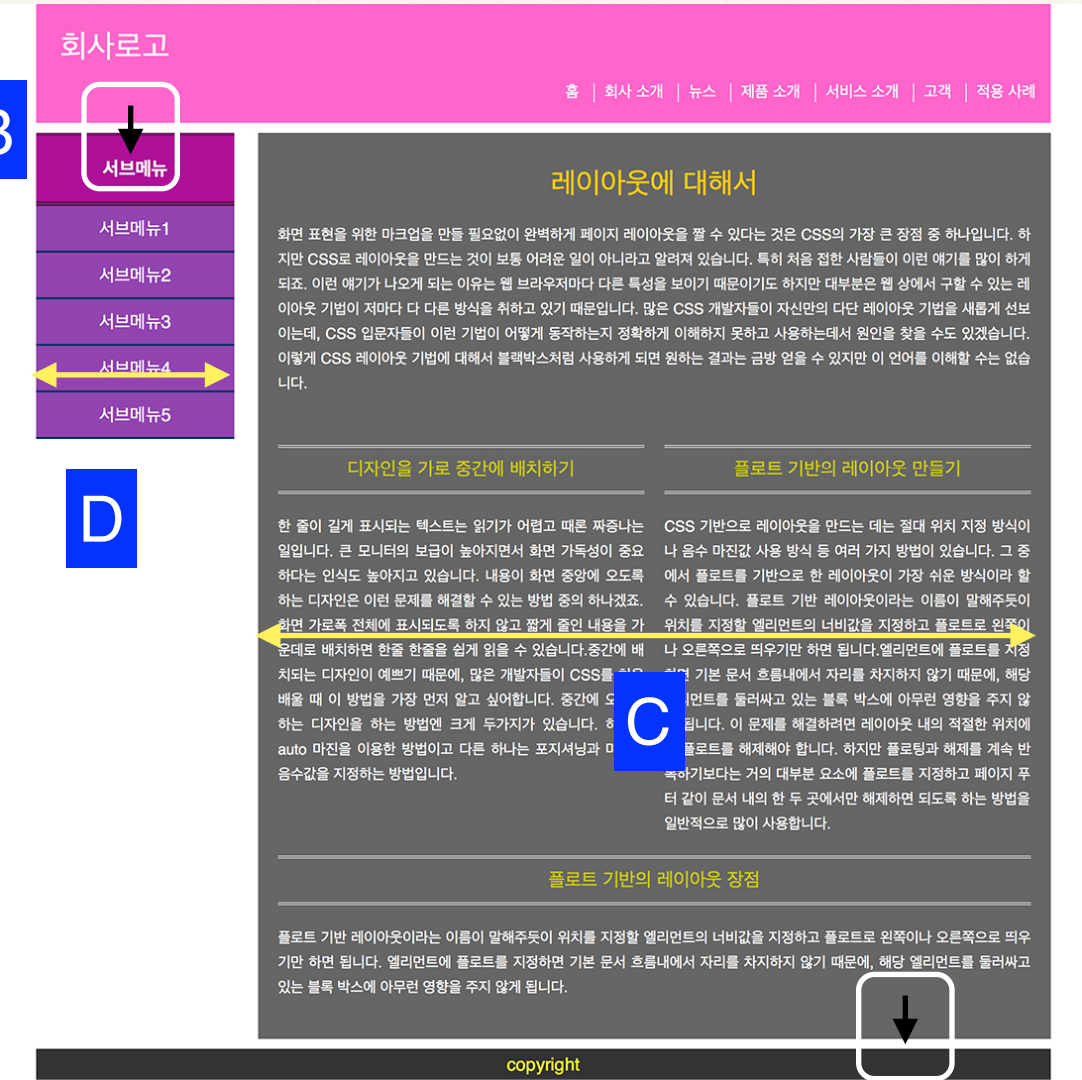
크기	가로	A
여백	가로중앙	

section

크기	가로	C
위치	오른쪽	
여백	아래	
바탕	칼라	

aside

크기	가로	D
위치	왼쪽	



footer

크기	가로	A
	세로	
font꾸미기	크기, 칼라, 정렬, 종류	
바탕	칼라	



header

여백	위 ,아래	A, B
----	-------	------

h1

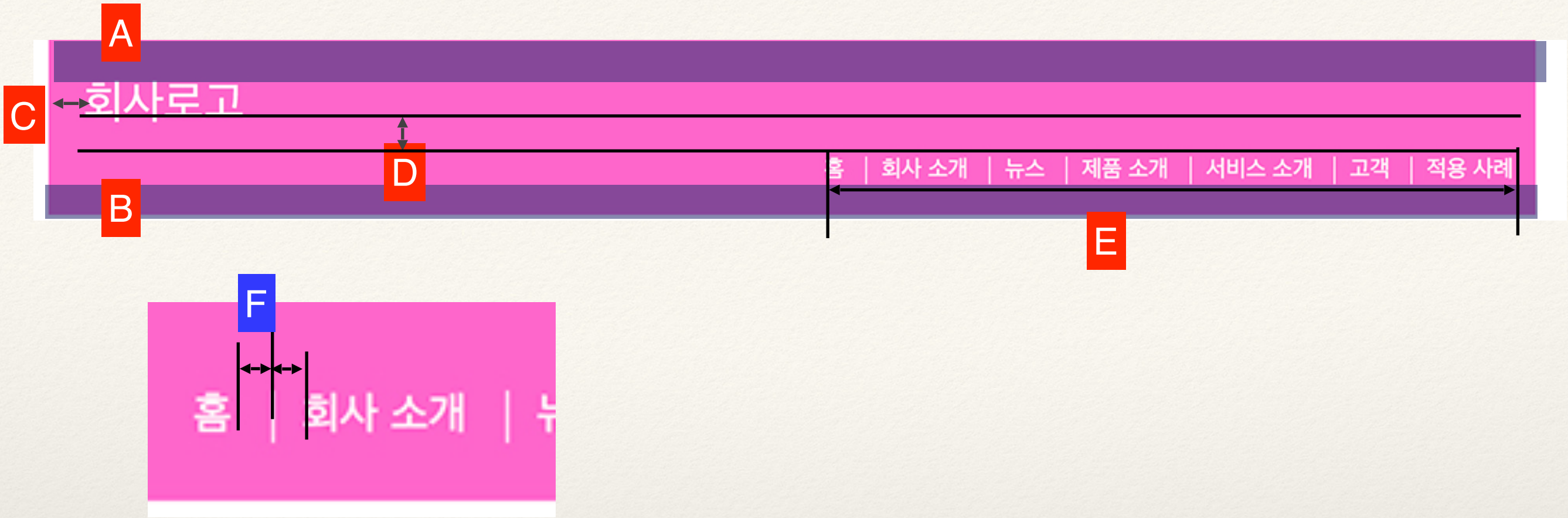
크기	가로	header크기
밖의 여백	아래	D
들여쓰기		C
font정리		

nav

크기	가로	E
위치	오른쪽	
	float 문제해결하기	

ul

가로	
----	--



li

성격변화	text처럼	
여백	오른쪽, 왼쪽	
보더	왼쪽	첫번째 li제외

a

font 정리		
---------	--	--

a:hover

font 칼라		
---------	--	--

section공통부분



화면 표현을 위한 마크업을 만들 필요없이 쉽게 페이지 레이아웃을 짤 수 있다는 것은 CSS의 가장 큰 장점 중 하나입니다. 하지만 CSS로 레이아웃을 만드는 것이 보통 어려운 일이 아니라고 알려져 있습니다. 특히 처음 접한 사람들이 이런 얘기를 많이 하게 되죠. 이런 얘기가 나오게 되는 이유는 웹 브라우저마다 다른 특성을 보이기 때문이기도 하지만 대부분은 웹 상에서 구할 수 있는 레이아웃 기법이 저마다 다 다른 방식을 취하고 있기 때문입니다. 많은 CSS 개발자들이 자신만의 다단 레이아웃 기법을 새롭게 선보이는데, CSS 입문자들이 이런 기법이 어떻게 동작하는지 정확하게 이해하지 못하고 사용하는데서 원인을 찾을 수도 있겠습니다. 이렇게 CSS 레이아웃 기법에 대해서 블랙박스처럼 사용하게 되면 원하는 결과는 금방 얻을 수 있지만 이 언어를 이해할 수는 없습니다.

디자인을 가로 중간에 배치하기

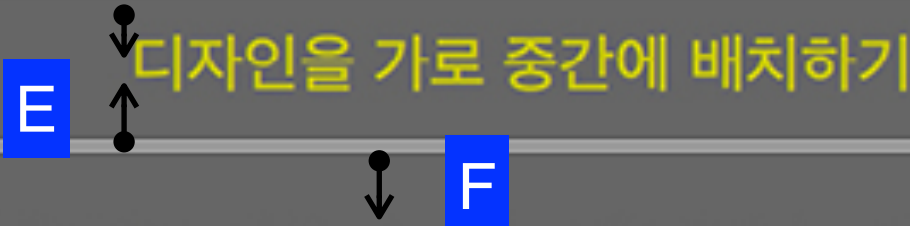
플로트 기반의 레이아웃 만들기

한 줄이 길게 표시되는 텍스트는 읽기가 어렵고 때론 짜증나는 일입니다. 큰 모니터의 보급이 높아지면서 화면 가독성이 중요하다는 인식도 높아지고 있습니다. 내용이 화면 중앙에 오도록 하는 디자인은 이런 문제를 해결할 수 있는 방법 중의 하나겠죠. 화면 가로폭 전체에 표시되도록 하지 않고 짧게 줄인 내용을 가운데로 배치하면 한줄 한줄을 쉽게 읽을 수 있습니다.중간에 배치되는 디자인이 예쁘기 때문에, 많은 개발자들이 CSS를 처음 배울 때 이 방법을 가장 먼저 알고 싶어합니다. 중간에 오도록 하는 디자인을 하는 방법엔 크게 두가지가 있습니다. 하나는 auto 마진을 이용한 방법이고 다른 하나는 포지셔닝과 마진에 음수값을 지정하는 방법입니다.

CSS 기반으로 레이아웃을 만드는 데는 절대 위치 지정 방식이나 음수 마진값 사용 방식 등 여러 가지 방법이 있습니다. 그중에서 플로트를 기반으로 한 레이아웃이 가장 쉬운 방식이라 할 수 있습니다. 플로트 기반 레이아웃이라는 이름이 말해주듯이 위치를 지정할 엘리먼트의 너비값을 지정하고 플로트로 왼쪽이나 오른쪽으로 띄우기만 하면 됩니다.엘리먼트에 플로트를 지정하면 기본 문서 흐름내에서 자리를 차지하지 않기 때문에, 해당 엘리먼트를 둘러싸고 있는 블록 박스에 아무런 영향을 주지 않게 됩니다. 이 문제를 해결하려면 레이아웃 내의 적절한 위치에서 플로트를 해제해야 합니다. 하지만 플로팅과 해제를 계속 반복하기보다는 거의 대부분 요소에 플로트를 지정하고 페이지 푸터 같이 문서 내의 한 두 곳에서만 해제하면 되도록 하는 방법을 일반적으로 많이 사용합니다.

플로트 기반의 레이아웃 장점

플로트 기반 레이아웃이라는 이름이 말해주듯이 위치를 지정할 엘리먼트의 너비값을 지정하고 플로트로 왼쪽이나 오른쪽으로 띄우기만 하면 됩니다. 엘리먼트에 플로트를 지정하면 기본 문서 흐름내에서 자리를 차지하지 않기 때문에, 해당 엘리먼트를 둘러싸고 있는 블록 박스에 아무런 영향을 주지 않게 됩니다.



section

여백정리	A,B,C	크기정리
------	-------	------

section>h3

크기		
여백	아래	D
font정리		
font정렬		

section h4

크기	부모에 맞추기	
여백(안)	위,아래	E
여백(밖)	아래	F
보더		
font 정리 정렬		

section p

크기	부모에 맞추기	
여백(밖)	아래	G
font정리 정렬		

#main 나머지 정리

레이아웃에 대해서

화면 표현을 위한 마크업을 만들 필요없이 완벽하게 페이지 레이아웃을 짤 수 있다는 것은 CSS의 가장 큰 장점 중 하나입니다. 하지만 CSS로 레이아웃을 만드는 것이 보통 어려운 일이 아니라고 알려져 있습니다. 특히 처음 접한 사람들이 이런 얘기를 많이 하게 되죠. 이런 얘기가 나오게 되는 이유는 웹 브라우저마다 다른 특성을 보이기 때문이기도 하지만 대부분은 웹 상에서 구할 수 있는 레이아웃 기법이 저마다 다 다른 방식을 취하고 있기 때문입니다. 많은 CSS 개발자들이 자신만의 다단 레이아웃 기법을 새롭게 선보이는데, CSS 입문자들이 이런 기법이 어떻게 동작하는지 정확하게 이해하지 못하고 사용하는데서 원인을 찾을 수도 있겠습니다. 이렇게 CSS 레이아웃 기법에 대해서 블랙박스처럼 사용하게 되면 원하는 결과는 금방 얻을 수 있지만 이 언어를 이해할 수는 없습니다.




A

디자인을 가로 중간에 배치하기

플로트 기반의 레이아웃 만들기

한 줄이 길게 표시되는 텍스트는 읽기가 어렵고 때론 짜증나는 일입니다. 큰 모니터의 보급이 높아지면서 화면 가독성이 중요하다는 인식도 높아지고 있습니다. 내용이 화면 중앙에 오도록 하는 디자인은 이런 문제를 해결할 수 있는 방법 중의 하나겠죠. 화면 가로폭 전체에 표시되도록 하지 않고 짧게 줄인 내용을 가운데로 배치하면 한줄 한줄을 쉽게 읽을 수 있습니다.중간에 배치되는 디자인이 예쁘기 때문에, 많은 개발자들이 CSS를 처음 배울 때 이 방법을 가장 먼저 알고 싶어합니다. 중간에 오도록 하는 디자인을 하는 방법엔 크게 두가지가 있습니다. 하나는 auto 마진을 이용한 방법이고 다른 하나는 포지셔닝과 마진에 음수값을 지정하는 방법입니다.



B

플로트 기반의 레이아웃 장점

플로트 기반 레이아웃이라는 이름이 말해주듯이 위치를 지정할 엘리먼트의 너비값을 지정하고 플로트로 왼쪽이나 오른쪽으로 띄우기만 하면 됩니다. 엘리먼트에 플로트를 지정하면 기본 문서 흐름내에서 자리를 차지하지 않기 때문에, 해당 엘리먼트를 둘러싸고 있는 블록 박스에 아무런 영향을 주지 않게 됩니다.

디자인을 가로 중간에 배치하기

한 줄이 길게 표시되는 텍스트는 읽기가 어렵고 때론 짜증나는 일입니다. 큰 모니터의 보급이 높아지면서 화면 가독성이 중요하다는 인식도 높아지고 있습니다. 내용이 화면 중앙에 오도록 하는 디자인은 이런 문제를 해결할 수 있는 방법 중의 하나겠죠. 화면 가로폭 전체에 표시되도록 하지 않고 짧게 줄인 내용을 가운데로 배치하면 한줄 한줄을 쉽게 읽을 수 있습니다.중간에 배치되는 디자인이 예쁘기 때문에, 많은 개발자들이 CSS를 처음 배울 때 이 방법을 가장 먼저 알고 싶어합니다. 중간에 오도록 하는 디자인을 하는 방법엔 크게 두가지가 있습니다. 하나는 auto 마진을 이용한 방법이고 다른 하나는 포지셔닝과 마진에 음수값을 지정하는 방법입니다.

section>p

여백정리	A	크기정리
------	---	------

article.left

크기	가로	
왼쪽 보내기		

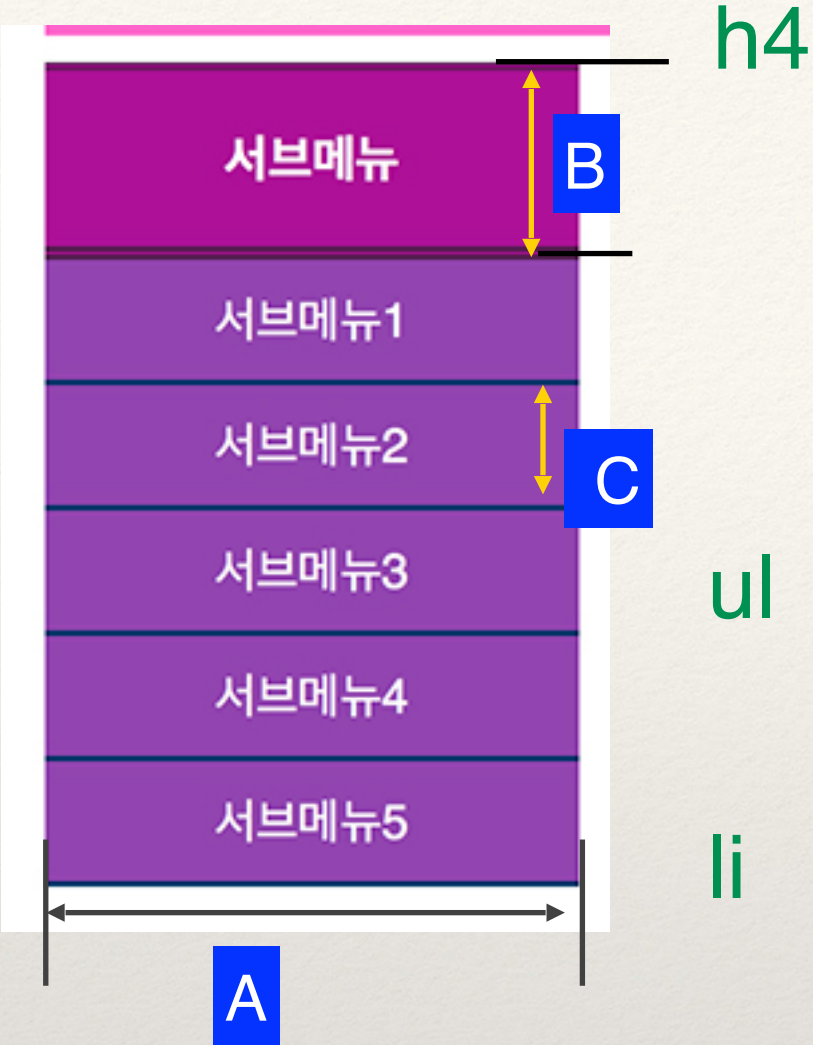
article.right

크기	가로	
오른쪽 보내기		

article.bottom

float해결		
---------	--	--

aside



크기	가로,세로	A,B
보더	위, 아래	
바탕	칼라	
font정리, 정렬		

크기	가로	A
----	----	---

크기	가로	ul
보더	아래	

성격변화		
크기	가로,세로	A,C
바탕	칼라	
font정리,정렬		

폰트칼라, 바탕		
----------	--	--